EP 501 Homework 5

Julio Guardado

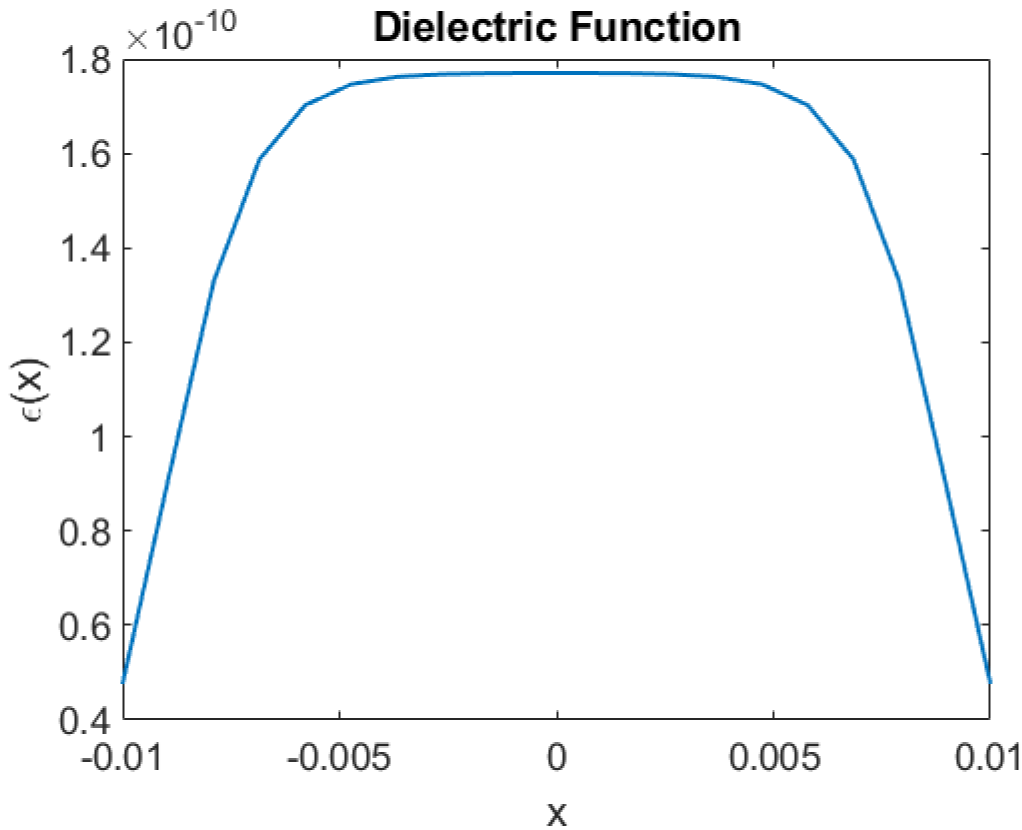
```
clc;clear;close all;
```

Problem 1 Setup

```
%define parameters
a = 0.01;           %m
l = a/5;            %m
x_prime = -9*a/10;  %m
x_dprime = 9*a/10;  %m
eps0 = 8.854e-12;

%define grid
lx = 20;
x = linspace(-a,a,lx);
dx = x(2)-x(1);
dp_dx = zeros(lx,1);
```

Problem 1a

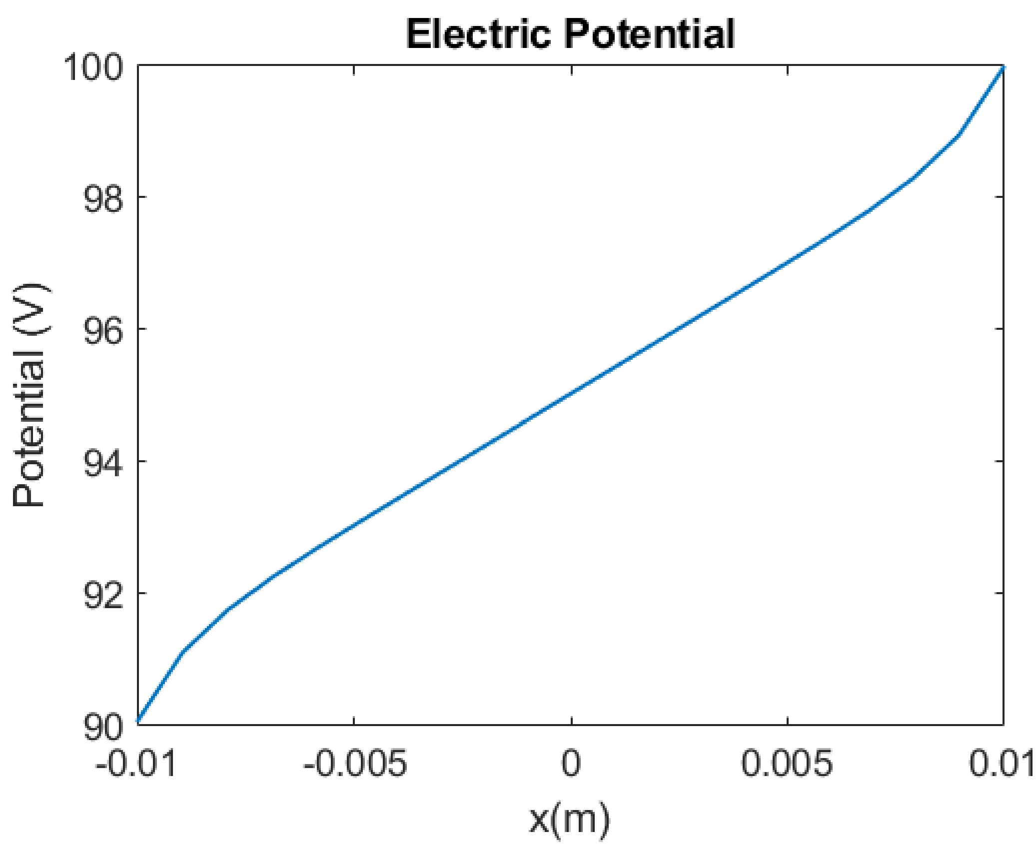
```
%calculate dielectric function
eps = eps0*(10*tanh((x-x_prime)/l)-10*tanh((x-x_dprime)/l));

%plot
figure(1)
plot(x,eps)
title('Dielectric Function')
xlabel('x'); ylabel('\epsilon(x)');
```



Problem 1b-d

```
%calculate derivative of epsilon
deps_dx = zeros(size(eps));

%forward diff for beginning
deps_dx(1) = (eps(2)-eps(1))/dx;

%take centered diff for middle
for i = 2:lx-1
    deps_dx(i) = (eps(i+1) - eps(i-1))/(2*dx);
end %for

%backward difference at end
deps_dx(lx) = (eps(lx)-eps(lx-1))/dx;

%make matrix
M = zeros(lx,lx);
b = zeros(lx,1);

%build matrix
M(1,1) = -1/dx;
M(1,2) = 1/dx;
M(lx,lx) = 1;
for ix = 2:lx-1
    %alpha
```

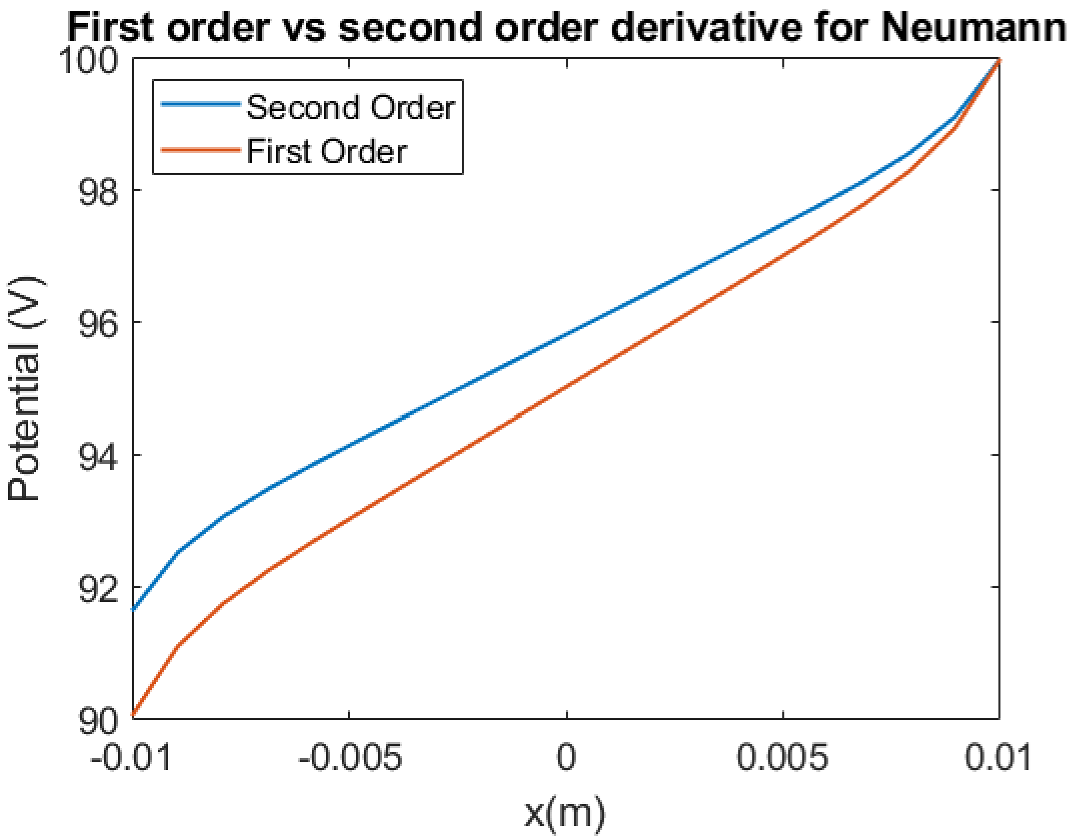
```
    M(ix,ix-1) = eps(ix)/dx^2 - deps_dx(ix)/(2*dx);
    %beta
    M(ix,ix) = -2*eps(ix)/dx^2;
    %gamma
    M(ix,ix+1) = eps(ix)/dx^2 + deps_dx(ix)/(2*dx);
end %for

%apply BCs
b(1) = 1000;
b(lx) = 100;

%solve
phi = M\b;

figure(2)
plot(x,phi)
title('Electric Potential')
xlabel('x(m)'); ylabel('Potential (V)')
```



Problem 1e

```
Mmod = M;
Mmod(1,3) = -1/2/dx;
Mmod(1,2) = 4/2/dx;
Mmod(1,1) = -3/2/dx;
```

```
phi_mod = Mmod\b;
figure(3);
plot(x,phi_mod,x,phi)
title('First order vs second order derivative for Neumann')
legend('Second Order','First Order','location','northwest')
xlabel('x(m)'); ylabel('Potential (V)')
```

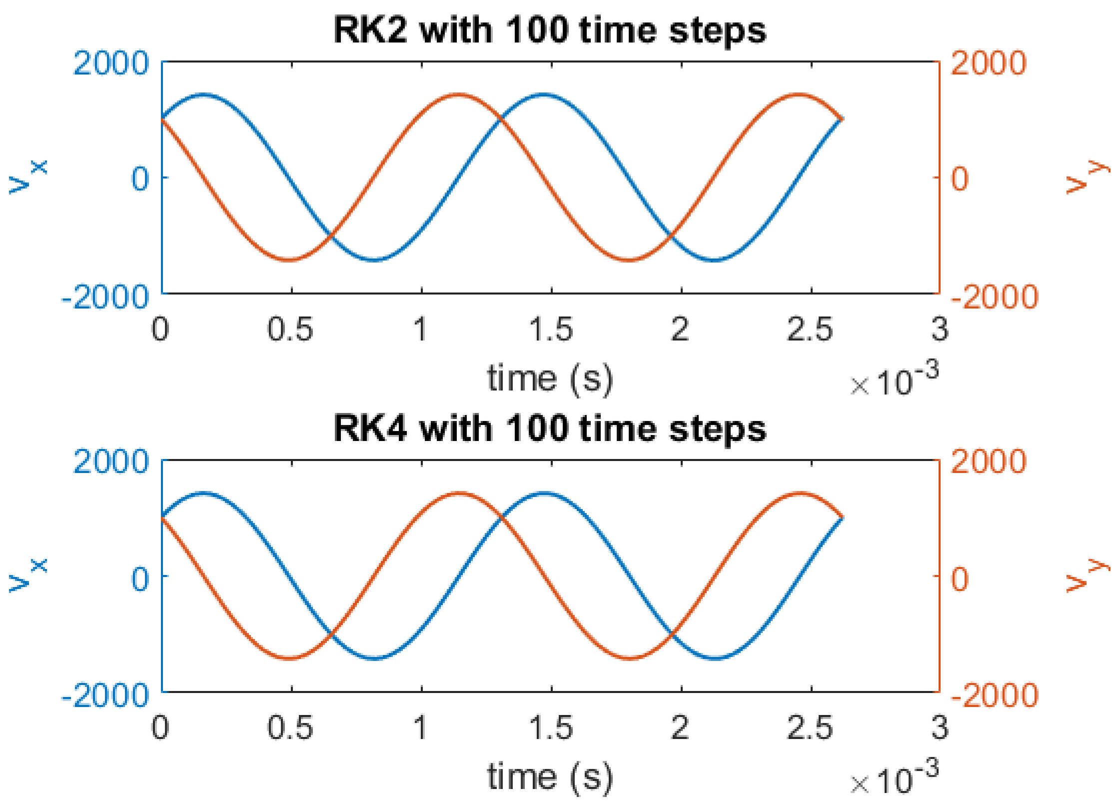


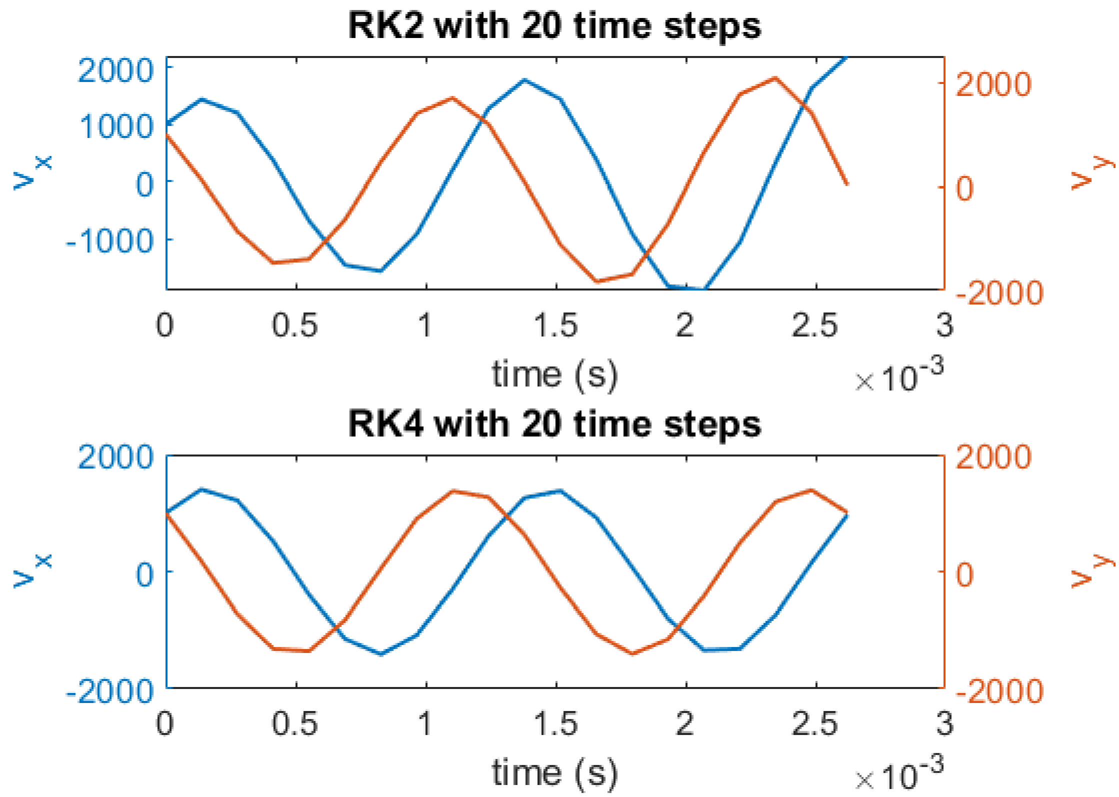
Problem 2

```
%set up
m = 1.67e-27;   %kg
q = 1.6e-19;    %C
B = 50000e-9;   %T

%particle oscillation period
omega = q*B/m;
T = 2*pi/omega;


%time grid
lt = 100;
t = linspace(0,2*T,lt);
dt = t(2)-t(1);

%rk2
```

```
[vx_rk2,vy_rk2] = rk2(1000,1000,t,omega);

%rk4
f = @(t,v) ([omega*v(2);-omega*v(1)]);
v_rk4 = rk4(t,[1000,1000],f);

figure(4)
subplot(2,1,1)
yyaxis left
plot(t,vx_rk2);
title('RK2 with 100 time steps')
xlabel('time (s)');
ylabel('v_x');

yyaxis right
plot(t,vy_rk2);
ylabel('v_y');

subplot(2,1,2)
yyaxis left
plot(t,v_rk4(1,:));
title('RK4 with 100 time steps')
xlabel('time (s)');
ylabel('v_x');

yyaxis right
plot(t,v_rk4(2,:));
ylabel('v_y');

%recalculate with less steps
ttt = linspace(0,2*T,20);

%rk2
[vx_rk2,vy_rk2] = rk2(1000,1000,ttt,omega);

%rk4
v_rk4 = rk4(ttt,[1000,1000],f);

figure(5)
subplot(2,1,1)
yyaxis left
plot(ttt,vx_rk2);
title('RK2 with 20 time steps')
xlabel('time (s)');
ylabel('v_x');

yyaxis right
plot(ttt,vy_rk2);
ylabel('v_y');

subplot(2,1,2)
yyaxis left
plot(ttt,v_rk4(1,:));
```

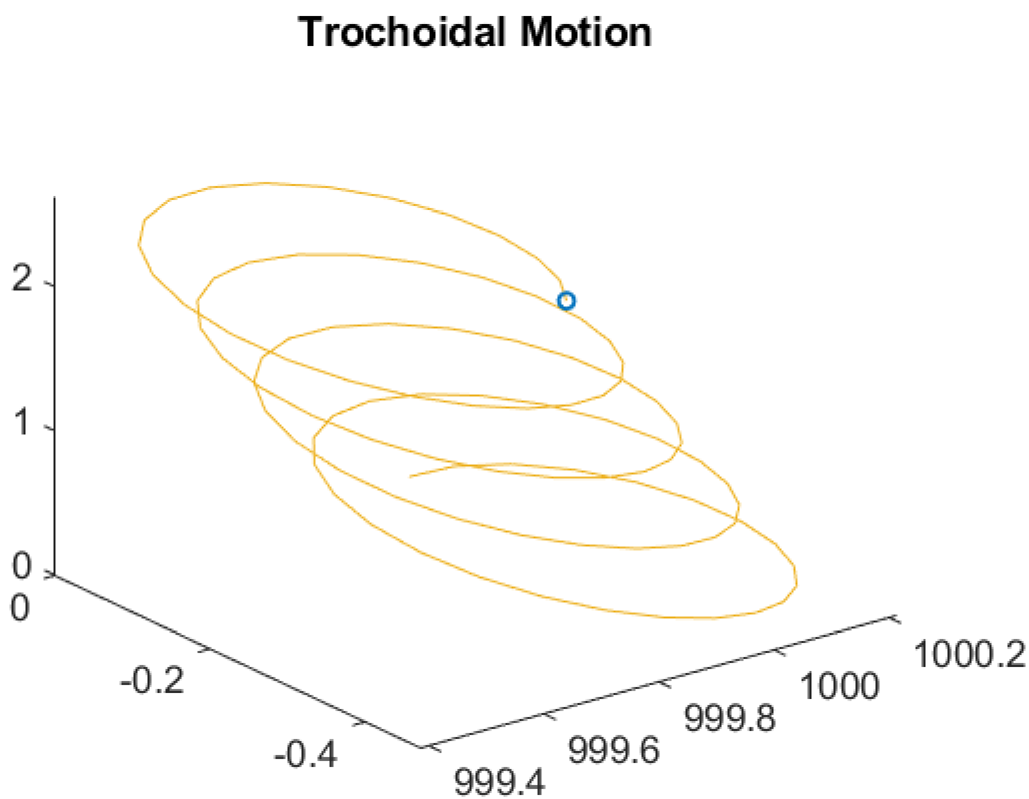
```
title('RK4 with 20 time steps')
xlabel('time (s)');
ylabel('v_x');

yyaxis right
plot(ttt,v_rk4(2,:));
ylabel('v_y');
```





Problem 2c

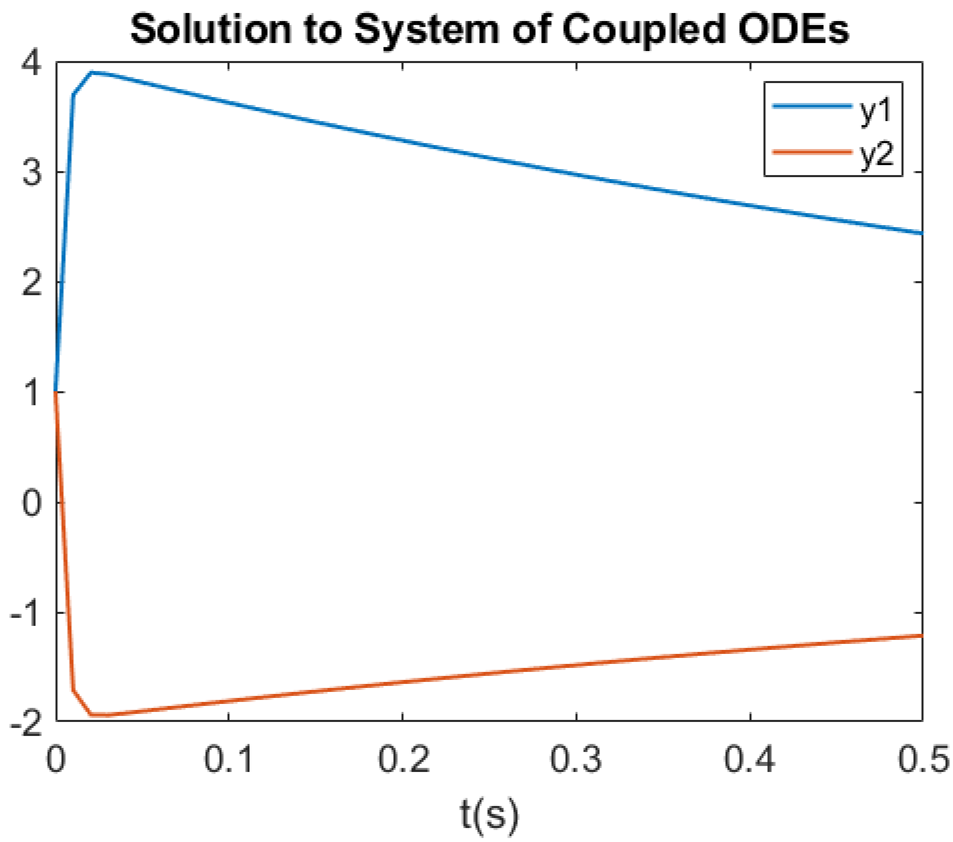
```
%initial conditions
alpha10 = 0;
beta10 = 0;
alpha20 = 1000;
beta20 = 1000;

%define function
f2 = @(t,v) ([v(2);-omega*v(4)*(1+.5*v(1));v(4);omega*v(2)*(1+.5*v(1))]);

%solve using rk4
t5 = linspace(0,5*T,100);
v_new = rk4(t5,[alpha10,beta10,alpha20,beta20],f2);

%calculate position
x = v_new(3,:);
y = v_new(1,:);
vz = 1e3;
z = vz*t;

figure(6);
comet3(x,y,z)
title('Trochoidal Motion')
```



Problem 3

```
tt = linspace(0,.5,50);
N = numel(tt);
dtt = tt(2)-tt(1);
y1=zeros(1,N);
y1(1)=1;
y2=zeros(1,N);
y2(1)=1;

% Decouple ODEs to get two first order ODEs using Jordan decomposition
% let v = -y1 - 2*y2, w = y1 + y2
% dv/dt = -1000*v, dw/dt = -w
v = 0.*y1;
w = 0.*y1;

%set initial conditions
v(1) = -y1(1)-2*y2(1);
w(1) = y1(1) + y2(1);

%use euler method to solve each ODE
for n=2:N
    v(n)=v(n-1)/(1+1000*dtt);
end %for

for n=2:N
```

```
    w(n)=w(n-1)/(1+dtt);
end %for

%go back to y1 and y2, two equations and two unknowns
y2 = -w-v;
y1 = w - y2;

figure(7);
plot(tt,y1,tt,y2)
title('Solution to System of Coupled ODEs')
xlabel('t(s)')
legend('y1','y2')
```



Published with MATLAB® R2019b